DEVICE PARAMETERS 1/

JPL PART #	MFR	GENERIC PART NO.	RADIATION LEVEL (TID) (RADS) 2/	PACKAGE STYLE	TERMINAL CONNECTIONS	ELECTRICAL PERFORMANCE CHARACTERISTICS	ELECTRICAL TEST REQUIREMENTS	BURN-IN CONNECTION TABLE
12159 - E01060FR	HONEYWELL SSEC	HR1060 - CDU	100K	FIG. 6-3 HEREIN (84-LEAD FLATPACK)	FIG. 6-1 HEREIN	TABLES 5-4 & 5-5 HEREIN	TABLE 5-1 HEREIN	TABLE 5-7 HEREIN

NOTES: 1/ THIS DRAWING, IN CONJUNCTION WITH CS515837B AND MIL-I-38535, LEVEL V, IMPOSES ALL REQUIREMENTS FOR PROCUREMENT OF THESE DEVICES.
2/ THE POST-IRRADIATION PARAMETRIC LIMITS SHALL BE THOSE OF TABLES 5-4 & 5-5 HEREIN.
3/ THIS STANDARD TAKES PRECEDENCE OVER DOCUMENTS REFERENCED HEREIN.

RELEASED THRU SECTION 356 DATA MANAGEMENT: DATE:

REVISION: C APPROVED BY (Section 514): DATE:

APPROVED BY (Section 336): DATE:

APPROVED SOURCE(S)

VENDOR PART NO.	VENDOR	JPL PART NO.
22016581	HONEYWELL INC. SOLID STATE ELECTRONICS CENTER PLYMOUTH, MN 55441 CAGE NO. 34168	12159 - E01060FR

The item listed in the approved source block and identified by vendor name, address, and part number will be evaluated and tested by the JPL Electronic Parts Reliability Section or its delegated alternate before being approved for use. Non-JPL users shall check with the Electronic Parts Reliability Section on the status of the part's approval before using.

JET PROPULSION LABORATORY CALIFORNIA INSTITUTE OF TECHNOLOGY CAGE NO. 23835

Procurement Specification: CS515837

Screening Specification: CS515837, ST12159

Custodian: Parts Reliability Group Section 514

TITLE:

MICROCIRCUIT, DIGITAL, ASIC, GATE ARRAY,
COMMAND DETECTOR UNIT (CDU)

DETAILED SPECIFICATION

ST 12159

FILE: (Sec.514): F:\USERS\514\SPECS\ACT-DETL\ST12159.C

CHANGE INCORPORATION LOG

CHG LTR	ENGINEERING APPROVAL			PAGES AFFECTED	RELEASED THRU		
	INITIAL	SECTION	DATE		INITIAL	SECTION	DATE
A	HS DH	514 336	8/24/92 8/27/92	Initial Release	RA	356	12/22/92
B	HS DH	514 336	3/16/93 3/16/93	1-7, 9-11, 21-24, 39-44, 47, 48, 52	RA	356	3/17/93
C	HS DH	514 336	5/20/94	2, 9-11, 39-41, 43, 45, 47-52			

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Scope

This is the detailed specification for a space-qualified signal processor gate array ASIC for the Command Detector Unit of the Radio Frequency Subsystem. This document shall be the sole source of design specifications for the CDU gate array, and shall supersede any other specification documents issued prior to this release.

Applicable Documents

- *General Specification for Gate Array Application Specific Integrated Circuits (ASICs), 24 March 1992, CS515837, Rev. B*

  This document establishes the general design system, manufacturing and testing requirements for the gate array Application Specific Integrated Circuit (ASIC) parts.

- *NASA Deep Space Command Detector Unit Development Final Engineering Report,* 9 June 1987, JPL D-4233, J.B. Berner.

  Primary source for algorithms and system analysis.

- *CDU Processor Chip Design Book,* Sec. 348 VLSI Design Group

  This document describes the detailed design of the CDU chip at the schematic level, the simulation strategies and results, and any pertinent analysis results. Continuously updated; not a general release document.

Conventions Used

In naming signals, a leading "n" in the name denotes an active-low signal, as in "nPOR".

Bus naming conventions follow Mentor, where Name(x:y) refers to all bits in the bus Name from x to y, inclusive, and Name(x,y,z) refers to bits x, y, and z in bus Name. Mentor does not support multidimensional or hierarchical buses, so names like Name(x:y)(n:m) or Name (Name2, Name3) are meaningless. The name written without subscripts is taken to mean the bus as a whole.

Highlights mark passages that have been altered, expanded, added, or whose status as options have changed since the previous document revision.

1. CHIP OVERVIEW

The Command Detector Unit (CDU) ASIC implements the digital segments of several phase-locked loops required to track subcarrier phase-modulated NRZ bit streams from the Cassini transponders and to provide the resulting digital clock and data bit streams to the Command and Data Subsystem of the spacecraft. The CDU also provides status to external devices, in particular, ground support equipment and on-spacecraft telemetry.

The block diagram in figure 1-1 shows the CDU ASIC in the context of the Command Detector Unit. The CDU implements mature, well-characterized algorithms developed over several flight missions. No substantive alteration to these algorithms is anticipated for the CDU subsystem. The algorithms are implemented using microcode stored in external ROM, providing flexibility in debugging and upgrading the CDU.

Figure 1-1 shows the boundary between the ASIC and the other components in the Command Detector Unit. In general, the ASIC will perform the digital operations previously conducted by an 80C86 microprocessor, together with its RAM, and IO devices. Design goals are higher performance, lower mass, lower power dissipation, and higher radiation tolerance than the Mars Observer implementation.

The ASIC bears some architectural resemblance to a typical Digital Signal Processor (DSP), and therefore may find application in other systems that have similar requirements for low-frequency analog sampling and synchronous serial output of processed data. However, no claim is made concerning the suitability of this part for any other application.

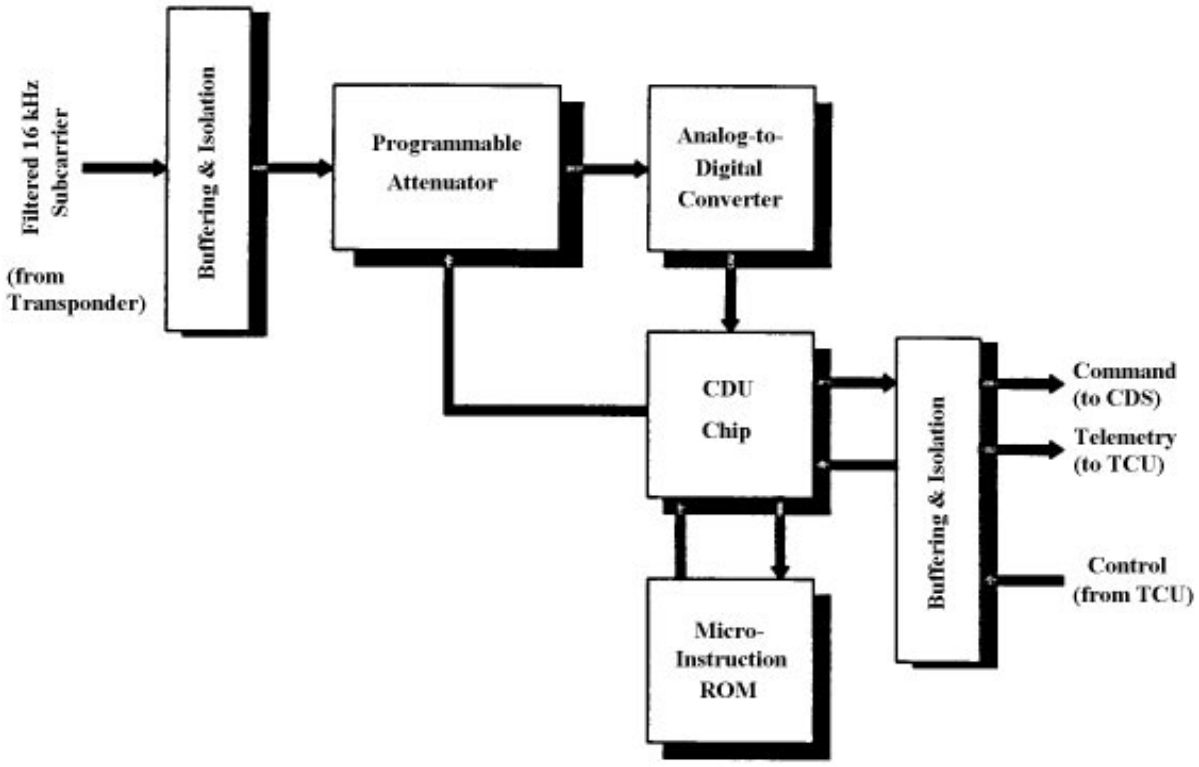


Figure 1-1. CDU ASIC in Command Detector Unit

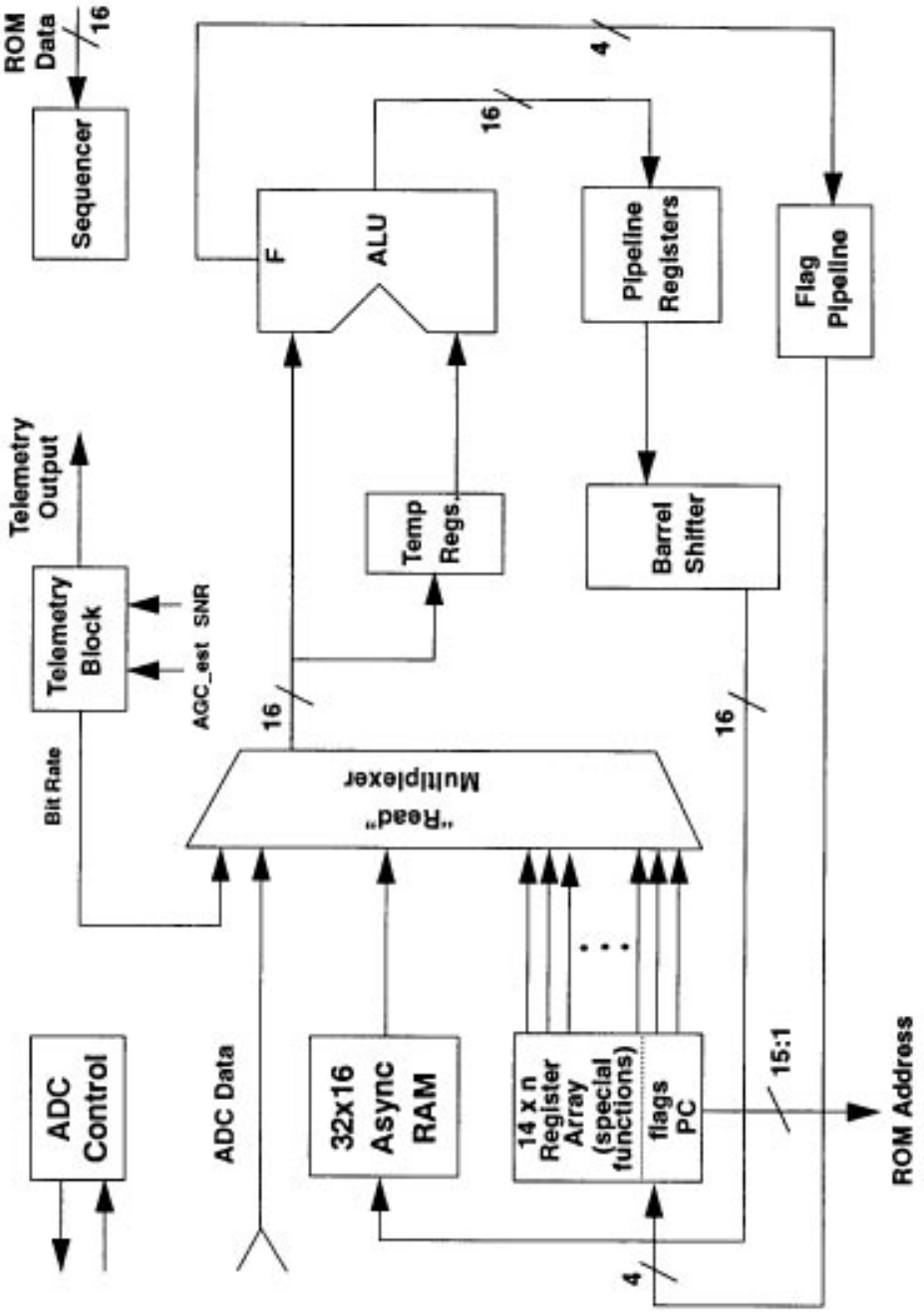


Figure 1-2. CDU ASIC Block Diagram

2. CDU ASIC SIGNAL DESIGNATIONS AND DESCRIPTIONS

2.1 CDU ASIC Symbol

Figure 2-1 shows the signal name designations of the ASIC, exclusive of power and ground. [1]

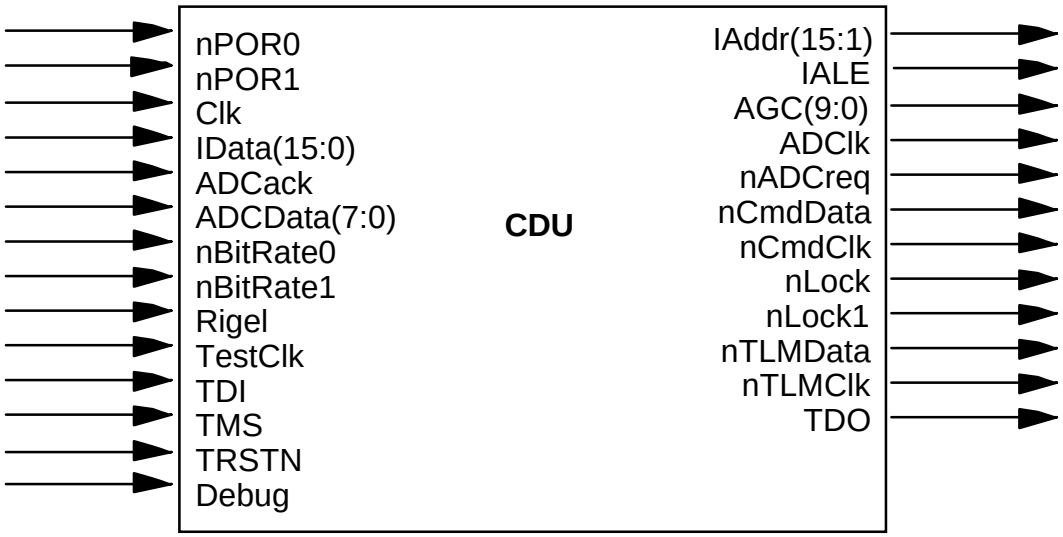


Figure 2-1. CDU ASIC Symbol

Table 2-1 lists the CDU Processor Chip signals and their descriptions. Actual pin assignments are specified in section 6. Pad locations on the die are to be assigned in conjunction with Honeywell during the layout phase in order to conform to the pin assignments.

Test modes that exist for microcode debugging or hardware test will alter the meaning and/or behavior of the pins. These test modes shall not be commanded or commandable in flight hardware; these modes will only be available for ground-based development applications.

The ASIC has 83 pin assignments, allowing a margin of one (1) pin for putting the part in an 84-pin package.

---

[1] Figure 2-1 is in the symbol format used by Mentor Graphics CAD packages. CDU_ASIC is the symbol name. I$00 is the placeholder for the instance property, that is analagous to a part reference number. These are artifacts of the data base implementation, and are not part of the ASIC specification.

2.2 CDU ASIC Signal Description

Name	Pin No.[1]	Type[2]	Drive[3]	Description
nPOR0 **nPOR1**	31 34	SINPU	AL	**Master Power-On Resets.** Forces all internal storage nodes to defined states. Will perform an asynchronous reset, except for specific nodes described explicitly elsewhere in the functional description. These signals are ANDed together; either going low will cause the reset.
Clk	24	IN	---	**Master Digital Clock.** Input from external fundamental crystal-controlled oscillator, from which all CDU timing is derived.
IAddr(15:1)	66 - 73, 76 - 82	OUT9	AH	**Instruction Address.** The word address into external microcode ROMs (Flashes internal data (15:1) in Debug mode).
IALE	65	OUT9	AH	**Instruction Address Latch Enable.** High when IAddr is changing values (Flashes internal data bit 0 in Debug mode).
IData(15:0)	62 - 55, 52 - 45	IN	AH	**Instruction Data.** The partial instruction returned by the instruction ROMs.
AGC (9:0)	14 - 23	OUT3	AH	**Automatic Gain Control output.** Unsigned magnitude for setting the gain of the amplification stage prior to the Sample & Hold circuit.
ADClk	3	OUT3	AH	**Analog-to-Digital conversion Clock.** A division of Clk, for driving the A/D converter chip.
nADCreq	2	OUT3	AL	**Analog-to-Digital Conversion request.** Commands the ADC chip to start conversion of the analog voltage to a digital value, and place it on the ADCData input bus. Held until the converted value is internally accumulated.
ADCack	83	IN	↑	**Analog-to-Digital Conversion acknowledge. A** rising edge signifies to the CDU ASIC that the ADC chip has completed its conversion.
ADCData(7:0)	13, 10 - 4	IN	AH	**Analog-to-Digital Converter Data.** Unsigned magnitude of sampled-and-held analog input voltage. Used for both data and error accumulations.
nBitRate0 **nBitRate1**	37 38	INPU	AL	**Bit Rate sample descriptor.** Selects number of sample accumulations per bit, ergo the effective bit rate. These are serial inputs, sampled on nTLMClk falling and ANDed together. See Bit Rate Register for proper interpretation of these inputs.
nCmdData	41	OUT3	AL	**Command Data Bit output.** The serial output stream of demodulated command bits, outputting the D flag of the FLAGS register.

Table 2-1. Signal Descriptions

1 The pin numbers of bus signals are in the same sequence as the signals.
2 For a description of the signal type refer to table 5-9.
3 Drive may be active high (AH), active low (AL), active high tristate (AHZ), etc.

Name	Pin No.	Type	Drive	Description
nCmdClk	42	OUT3	↓	**Command Data Bit Clock.** When CDU is in lock, a falling edge on this output marks a valid bit value on CmdData.
nLock	36	OUT3	AL	**Subcarrier Lock Status.** Signifies that CDU is in lock. Outputs the inverse of L flag of the FLAGS register.
nLock1	35	OUT3	AL	**Subcarrier Lock1 Status.** Signifies that CDU has detected a subcarrier above lock threshold. Outputs the inverse of L1 flag of the FLAGS register.
nTLMData	39	OUT3	AL	**Telemetry Output Data.** Serial bit stream from internal telemetry registers.
nTLMClk	40	OUT3	↓	**Telemetry Output Clock.** Free-running 64 kHz clock derived from system clock. The falling edge of this clock marks valid data bits on TLMData, and the sampling of inputs nBitRate0 and nBitRate1.
Rigel[4]	25	INPD	AH	**Rigel test command.** Forces internal logic to assume functionality compatible with the Rigel test vector generation tool.
TestClk[4]	26	INPD	↑	**Test Circuitry Clock.** When CDU is being tested using scan path logic, this is the clock for test logic.
TDI[4]	28	INPU	AH	**Test Data In.** Serial data in for scan path test logic.
TMS[4]	27	INPU	AH	**Test Mode Select.** Commands the CDU chip scan logic mode.
TRSTN[4]	30	SIN	AL	**Test Reset.** Forces all nodes visible to the scan paths to a known state for test purposes. This was not required to be the same as nPOR(0, 1), but for simplicity's sake, we have defined it to perform identically to nPOR(0, 1).
TDO[4]	29	TRI3	AHZ	**Test Output Data.** Serial bit stream from the internal scan paths. High-impedance state controlled by OCM internal test logic.
Debug	63	IN	AH	**Debug Enable.** Forces CDU chip into software debug mode, enabling special instruction modes, suppressing CDU ROM buffers, suppressing Program Counter increments, etc.
VddP(5:0)	75, 53, 44, 33, 11, 1	VDD	---	**Vdd Power Pads.**
VssP(5:0)	74, 54, 43, 32, 12, 84	VSS	---	**Vss Power Pads.**

Table 2-1. Signal Descriptions (cont'd)

[4] These signals are defined for use with the Honeywell On-Chip Monitor (OCM) scan path control block, and their Rigel test vector generator. See Honeywell manuals for precise definitions of these signals. These signals are not expected to be used in flight hardware. See other OCM signals.

3. FUNCTIONAL DESCRIPTION

The CDU chip will implement tasks that fall into two categories: autonomic and programmed. Autonomic functions will consist of functions that are maintained in hardware transparent to the programmer. Autonomic functions will proceed in the absence of program interaction. Programmed functions will be implemented by an instruction sequencer internal to the CDU chip, fetching the instruction stream from an external ROM.

The CDU ASIC programming model is presented in detail sufficient for the reader to produce the required system functions and to predict their performance.

3.1 Autonomic Functions

The autonomic (background) functions in the CDU chip are intended to support the sampling and accumulation at two points of the incoming modulated subcarrier, on every subcarrier cycle; nominally, the first sample is to be taken at the zero crossing of the subcarrier, and the second 90 degrees later in the cycle. These two sample points are required to be accumulated separately to implement Mars Observer algorithms.

The previous version of the CDU sampled on alternate subcarrier cycles due to limited program execution time. The following definitions for background hardware are capable of supporting either the intended (every cycle) or previous (alternating cycles) sample rates, under program control.

3.1.1 Analog-to-Digital Conversion Clock Counter

3.1.1.1 A counter will exist for supplying a 50% duty-cycle clock to the ADClk output pin. It will perform a 1/4 division of the system clock, so as to supply an output period of 2.048 MHz when Clk is given 8.192 MHz.

3.1.1.2 On Power-On Reset, the counter will be forced to a known state. The ADClk bit will go to a logic 0 value, and remain at that value until the POR request terminates.[5]

3.1.2 Sample Counter

3.1.2.1 An 10-bit programmable counter will exist, known as the Sample Counter (SCount). It will be used to mark the several phases of the subcarrier sampling and accumulation cycle. In other words, it will govern the timing of sample requests to the external Analog-to-Digital Converter. A complete cycle of the phases is defined as follows:

1. At the beginning of this phase, the nADCreq signal to the external A/D converter will go active, the Sample Counter will load with the length of this phase from the SCntPh0 register, and an accumulation into EACC will be scheduled.

2. At the beginning of this phase, the nADCreq signal to the external A/D converter will go active, the Sample Counter will load with the length of this phase from the SCntPh1 register, and an accumulation into DACC will be scheduled.

3. The final phase is reserved for adjustment of the complete sample cycle. Under normal circumstances, the length of the final phase shall be loaded from the SCntPh2 register. If, however, the SCntBump register has

[5] The POR request can be initiated from any of the pins nPOR0, nPOR1, or TRSTN.

been written by software since the Sample Counter was last at this point in the cycle, the length of the final period will be loaded from SCntBump. The value in SCntBump will only be used once; subsequent adjustments to the overall period require another software write to SCntBump.

In sum, the values set in SCntPh0 through SCntPh2 set a basic sample cycle length. SCntBump may be used to make temporary adjustments to the overall period. SCntPh0 sets a constant time interval between the samples accumulated into EACC and those into DACC. SCntPh1 sets a constant time period necessary for the microcode to calculate any necessary adjustments to the overall cycle period. SCntPh2 contains the nominal length of the final phase. Adjustment to the total cycle length is made by writing to the SCntBump register before the final phase. If such a write to SCntBump occurs, SCntBump will substitute for SCntPh2 for that one cycle.

3.1.2.2 The Sample Counter will count at the frequency of the Clk input signal. As a result of the clocking frequency being much higher than the instruction execution frequency, this sample counter cannot be directly readable, writable, or clearable by program control.[6]

3.1.2.3 Given a Clk input of >8 MHz, and a sample cycle frequency of 16 kHz, the timing margins of the software algorithms require a minimum number of bits per phase. The calculated minimum bits per phase register are:

SCntPh0	8
SCntPhl	10
SCntPh2	8
SCntBump	8

When values of resolution less than the Sample Counter (10 bits) are stored, they will be automatically 0 - extended by hardware to the full counter width before being loaded into the counter.

3.1.2.4 The Sample Counter period arguments, as stored in the phase registers, shall represent (n+1) clock cycles of actual time. In other words, a value of 0 in SCntPh0 will result in a 1-cycle phase 0 before loading SCntPh1. No value will result in either 0 or infinite duration phases.

3.1.3 Sample Accumulation

3.1.3.1 Two "autonomic" 16-bit accumulators will exist in the design. One is known as the Error Accumulator (EACC), and the other as the Data Accumulator (DACC). Accumulations will occur, using ADC values, as a background task.

3.1.3.2 The Sample Counter (above) initiates the EACC and DACC accumulations. This includes initiating analog-to-digital conversions with the nADCreq pin. Latency of the A/D converter shall not affect the length of a Sample Counter phase.

3.1.3.3 If a requested A/D conversion completes in Sample Count Phase 0 (rising edge of ADCack), the CDU ASIC will accumulate the ADC value into EACC as follows:

$$EACC = EACC + ((ADCData + \$FC00) \bmod 2^{16})$$

3.1.3.4 If a requested A/D conversion completes outside of Sample Count Phase 0 (rising edge of ADCack), the CDU chip will accumulate the ADC value into DACC as follows:

---

[6] The reprogramming of a phase register at an incorrect time may corrupt the count sequence. It is the responsibility of the software to avoid such conflicts.

DACC = DACC +((ADCData + $FC00) mod 2^{16})

3.1.3.5 Transitions on ADCack that are not a direct result of the Sample Counter making a request on Convert will be ignored by accumulation hardware. This prevents noise on either nADCreq or ADCack from causing an accumulation at an inappropriate time.

3.1.3.6 The appropriate EACC or DACC accumulation will occur during the next instruction period after the conversion is finished. In other words, latency from conversion complete to accumulation complete shall be no more than one instruction period.

3.1.3.7 Neither the EACC or DACC accumulators will be allowed to overflow due to a sample accumulation; when an overflow condition would occur, the accumulators will assume the maximum value of the proper sign. This occurrence will not be flagged.

3.1.3.8 A new request for conversion to the ADC is not permitted until the previous conversion has been accumulated. This sets constraints on the programming of the Sample Counter phases.

3.1.3.9 The accumulation timing will be such that it will not interfere with software attempts to utilize an accumulated value; i.e., no conflicts will occur due to any attempts by software to read or write the accumulators.

3.1.3.10 The accumulators will be readable as instruction operands, and writable under software control. The act of reading either accumulator shall not alter its contents.

3.1.4 Bit Accumulation Counter

3.1.4.1 An 11-bit unsigned counter (BACnt) will exist to count the number of "accumulation cycles," here defined as the number of EACC/DACC accumulation pairs that have occurred. Since a DACC should never occur without a preceding EACC, it is deemed sufficient for the counter to count only DACC events.

3.1.4.2 The Bit Accumulation counter will be directly readable, or writeable by program control. Hardware will coordinate read and writes such that hardware conflicts will not occur between instruction operations and the increment count activity.

3.1.4.3 The Bit Accumulation counter will mark three phases (similar to the Sample Counter). Unlike the Sample Counter the Bit Accumulation counter will not initiate or govern external activities. Its purpose is to flag meaningful points in the accumulation background function for higher-level software algorithms.

1. At the beginning of the 0 phase, the B flag in the FLAGS register will be cleared to 0 to mark the beginning of a Bit Accumulation cycle. The Bit Accumulation Counter will load with the length of the first phase from the BACntPh0 register.

2. At the beginning of the 1 phase, the B flag in the FLAGS register will be set to 1 to mark the middle of a Bit Accumulation cycle. The Bit Accumulation Counter will load with the length of the first phase from the BACntPh1 register.

3. The final phase is reserved for adjustment of the complete Bit Accumulation cycle. Under normal circumstances, the length of the final phase shall be loaded from the BACntPh2 register. If, however, the BACntBump register has been written by software since the Bit Accumulation Counter was last at this point in the cycle, the length of the final period will be loaded from BACntBump. The value in BACntBump will

only be used once; subsequent adjustments to the overall period require another software write to BACntBump.

3.1.4.4 Given a sample cycle frequency of 16 kHz, the timing margins of the software algorithms require a minimum number of bits per phase. The calculated minimum bits per phase register are:

BACntPh0	11
BACntPhl	10
BACntPh2	11
BACntBump	11

When values of resolution less than the Bit Accumulation Counter (11 bits) are stored, they will be automatically 0-extended by hardware to the full counter width before being loaded into the counter.

3.1.4.5 The Bit Accumulation Counter phase lengths, as stored in the phase registers, shall represent (n+l) clock cycles of actual time. In other words, a value of 0 in BACntPh0 will result in a 1-cycle-long phase before loading BACntPh1. No value will result in either 0 or infinite-duration phases.

3.1.5 Telemetry Control Unit Interface

The CDU chip provides facilities for communicating with the Telemetry Control Unit (TCU).[7] The interface consists of three functions: Power-On Reset, Bit Rate Control, and Telemetry Output. This specification assumes that buffering or other level translation external to the CDU ASIC is non-inverting logic.

3.1.5.1 Power-On Reset

3.1.5.1.1 The Power-On Reset function shall asynchronously force all ASIC internal nodes and external pins into a known (i. e., predictable or deterministic) state. This implies that all ASIC outputs will seek a DC state, including clock outputs.

3.1.5.1.2 Power-On Reset shall be delivered on three active-low inputs nPOR0, nPOR1, and TRSTN; any input going active will cause the POR state to be entered. External buffering and cabling should therefore be designed such that a stuck-at-active fault is not possible.

3.1.5.1.3 Control signal outputs shall adopt an inactive value when the CDU ASIC is in the POR state. In particular,

IALE	0
ADClk	0
nADCreq	1

[7] Specifications here reflect JPL memo 3366-91-303.

nCmdClk	1
nTLMClk	1
nLock	1
nLock1	1

Other outputs are not considered control outputs, and may take any state without hazard. POR states of other outputs are defined within their specific functional descriptions elsewhere in this document.

3.1.5.2 Bit Rate Inputs

3.1.5.2.1 The CDU bit rate argument will be set by a bit-serial NRZ synchronous input word from the TCU. One received serial word will consist of one start bit, 4 data bits, and no stop bits. The quiescent state of the line shall be 1, and the start bit shall be 0. Data bits shall be "inverted" values, in recognition of the active-low character of the input lines. CDU microcode will interpret the meaning of the data field to set the actual bit rate.

3.1.5.2.2 The bit values of the Bit Rate input pin will be sensed on or near the falling edge of nTLMClk. Minimum setup and hold time about this reference shall be 50ns.[8]

3.1.5.2.3 The Bit Rate word may be delivered on either of two active-low inputs, nBitRate0 and nBitRate1; either input, or both, going active (low) will represent a logic 1 bit to the internal input shifter. External buffering and cabling should therefore be designed such that a stuck-at-active fault is not possible. Also, if Bit Rate words are received on both inputs, they must agree over the setup and hold regions about the sense point.

3.1.5.2.4 The Bit Rate word shall be received only once after a CDU Power-On Reset; further transitions of nBitRate(1, 0) shall be ignored by the ASIC until the new POR has occurred.

3.1.5.2.5 The bits of the Bit Rate word shall be defined as follows:

Bit #	Description	Comments
4:	BitRate(3)	
↓	↓	
1:	BitRate(0)	
0:	Start Bit	(Always 0)

3.1.5.3 Telemetry Output

3.1.5.3.1 A bit-serial NRZ synchronous output pin TLMData will provide telemetry information to the TCU. The transmitted telemetry word consists of one start bit, 32 data bits, and one stop bit. The quiescent state of the line shall be 0, and the start and stop bits shall each be 1. Data bits shall be inverted (active-low) values.

[8] Preliminary, pending further analysis of CDU/TCU interconnect.

3.1.5.3.2 The bit values of the telemetry output word may be sensed on or near the rising edge of nTLMClk. Minimum setup and hold about this reference shall be 100ns.

3.1.5.3.3 The telemetry word shall be sent under CDU instruction control. This allows for adjustment of the transmission interval to meet TCU requirements, without addition of handshaking logic, by simply altering the CDU instruction sequence.

3.1.5.3.4 The bits of the telemetry word shall be defined as follows:

Bit #	Description	Comments
33:	Stop Bit	(Always 0)
32:	AGC Estimate(7)	(l's compl of internal value)
↓	↓	
25:	AGC Estimate(0)	
24:	SNR Sample(15)	(l's compl of internal value)
↓	↓	
9:	SNR Sample(0)	
8:	Spare(1)	(Always 1)
↓	↓	
7:	Spare(0)	
6:	Lock1 Bit	(0 ⇒ CDU approaching Lock)
5:	Lock Bit	(0 ⇒ CDU In Lock)
4:	BitRate(3)	
↓	↓	
1:	BitRate(0)	
0:	Start Bit	(Always 0)

3.1.5.4 Telemetry Clock

A free-running clock output nTLMClk will run at approximately 64 kHz, derived from the ASIC master clock, Clk. The falling edge of this clock shall mark the point where both nTLMData and nBitRate(0, 1) are expected to be valid.

3.2 Programming Model

3.2.1 Address Map

The address map will consist of two uniform and independent address spaces: the program space and the operand space.

3.2.1.1 Program Address Space

3.2.1.1.1 The program space resides in external ROM; the pins IAddr(15:1) will provide a word address to the ROM, and IData(15:0) are the data returned from the ROM.

3.2.1.1.2 Instructions within the ROM will be organized as 32-bit words; all instructions shall be this length, and shall be stored in order of ascending address from LSW to MSW. i.e., even addresses access the LSW of instructions, while MSWs are on odd addresses.

3.2.1.1.3 Microcode execution after a Power-On Reset will commence at address $0000.

3.2.1.1.4 All instructions will be completely read in two fetches from ROM, regardless of operand type.

3.2.1.2 Operand Address Space

The instruction set supports operand references to internal devices and immediate ROM data. External RAM or other devices are not supported.

3.2.1.2.1 Values within the operand space shall be 16 bits in size. When physical ports, devices, or functions are less than this size, they shall be 0-extended internally such that all bits in the 16-bit operand are defined.

3.2.1.2.2 All writeable locations may be used as destination addresses for arithmetic operations.

3.2.1.2.3 The maximum size of the operand space shall be 64 words.

3.2.1.2.4 Address locations in table 3-1 marked as UNDEFINED are locations that perform no storage. Reads from these locations shall return $0000. Writes will have no effect, and are not considered errors.

Address	Description 15 0	Notes
$00:	EACC	(EACC and DACC are stored in RAM, if possible.)
$01:	DACC	
$02:	Bit Accumulation Cntr	
$03:	User RAM Space	
↓	↓	
$lF:	User RAM Space	
$20:	Bit Rate Input	($20 - $2F are read-only locations. Writes to
$21:	A/D Converter Input	these addresses will be treated as NO-OPs.)
$22:	UNDEFINED ($00)	(Reads from undefined locations return $00.)
↓	↓	
$2F:	UNDEFINED ($00)	
$30:	TLM.Stat	
$31:	TIM.AGCEst	
$32:	TLM.SNR	
$33:	UNDEFINED ($00)	
$34:	Sample Cntr Phase0	
$35:	Sample Cntr Phasel	
$36:	Sample Cntr Phase2	
$37:	Sample Cntr Bump	
$38:	Bit Accum Cntr Phase0	
$39:	Bit Accum Cntr Phasel	
$3A:	Bit Accum Cntr Phase2	
$3B:	Bit Accum Cntr Bump	
$3C:	UNDEFINED ($00)	
$3D:	AGC Control Word	
$3E:	FLAGS	
$3F:	Program Counter	

Table 3-1. Operand Address Map

3.2.2 Register Descriptions

3.2.2.1 Bit Rate Register (BRate)

15 8	7 0

N	0	0	0	0	0	0	0	0	0	0	0	RATE(3:0)			

The BRate register contains the inverse of the rate argument shifted in from the nBitRate(1, 0) input pins.

3.2.2.1.1 The N flag in the BRate register shall become 1 at Power-On Reset, marking the rate data as NOT VALID. Should an acceptable bit sequence appear at the nBitRate(1, 0) input pins, the N flag will be cleared to 0.

3.2.2.1.2 The Rate field of the BRate register shall be set to a value of $0 on Power-On Reset. This, with the preceding paragraph, yield $8000 as the value of the complete register at reset.

3.2.2.1.3 The BRate register is not software alterable. However, writes are not considered errors, as a wait loop can then look like this microcode example:

```
MOV     F, T, BRate, BRate      ; Set ALU flags according to BRate contents
SUB     NF, S, PC, $0002        ; Loop until N gets cleared.
```

This example does not take into account a timeout escape from the loop. A timeout operation might be required in actual microcode, should a system fault prevent reception of a valid bit rate sequence on nBitRate(1,0).

3.2.2.2 Flag Word (FLAGS)

15 8	7 0

T	C	V	S	Z	B	L	L1	D	U6	U5	U4	U3	U2	U1	U0

The FLAGS register contains all ALU and autonomic function flags; these coordinate software with hardware-detected events. Flags not used for hardware functions are available as user flags for coordinating software activities. The flags L, L1, D, and U0-6 are fully writeable by the instruction set in all modes of operation. The rest are hardware flags that are typically not directly writeable by the instruction set, except in Debug mode. For example, attempting to set the C flag by performing the instruction:

```
OR    F, T, FLAGS, #$4000       ; Attempt to set carry flag will not work in normal operation.
```

The software debug mode, controlled by the Debug input pin, allows for direct writing of the flag word, including hardware flags, by instructions issued from a debug monitor. This mode is not available when the CDU ASIC is installed in flight hardware. See section 4.2 for more complete information.

3.2.2.2.1 The bits of the FLAGS register are defined as follows:

Flag	Meaning
T	True -- Always a logic 1
C	Carry (from ALU)
V	Overflow (from ALU)
S	Sign (from ALU)
Z	Zero (from ALU)
B	Bit Accumulation Flag (Autonomic, from Bit Accumulation Counter)
L	Lock Flag (Software Output to nLock pin)
L1	Lock1 Flag (Software Output to nLock1 pin)
D	Data Flag (Software Output to nCmdData pin)
U6	User Flag 6
.	.
.	.
U0	User Flag 0

3.2.2.2.2 The ALU flags in the above table are set as a result of instruction execution. The instruction set documentation describes the calculations that result in new ALU flag values.

3.2.2.2.3 The B flag function is described in section 3.1.4 above. The inverse of the B flag value will be repeated at the nCmdClk output pin.

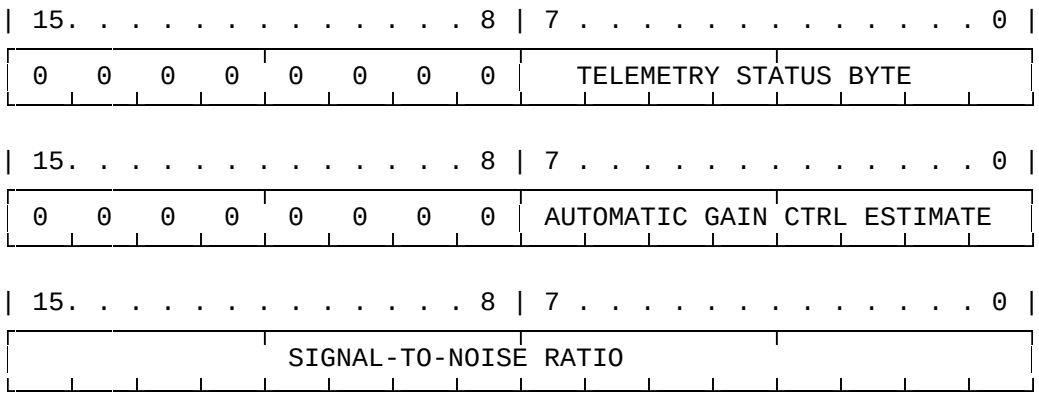
3.2.2.2.4 The flags L, L1, and D directly control output pins nLock, nLockl, and nCmdData, respectively. In all other respects they behave as user flags. A logic value 1 in these bits represents the active state of the controlled output pin.

3.2.2.2.5 The L flag is included as bit 5 in the telemetry output word.

3.2.2.2.5.1 The L1 flag is included as bit 6 in the telemetry output word.

3.2.2.2.6 The flag word will contain a value of %1000 0000 0000 0000 after a power-on reset.

3.2.2.3 Telemetry Registers (TLM.Stat, TLM.AGCEst, TLM.SNRSample)



3.2.2.3.1 These registers contain status reported by the telemetry function. These registers will be dumped into a shift register chain to perform the Telemetry Output function. The shifter operation to perform the Telemetry Output will not effect the values in these registers.

The Gain Control Estimate and Signal-to-Noise Ratio registers are written by software for the purposes of analyzing signal processing response. The Telemetry Status Register (TLM.Stat) is written by the software as well; it reports the "software" bit rate, i.e., the bit rate that software is actually attempting to decode, as opposed to the argument in the Bit Rate Register. Having software write this value provides more layers of checking of actual CDU operation.

3.2.2.3.2 The bit assignments in the TLM.Stat register are as follows:

15 8	7 0
0 0 0 0 0 0 0 0	0 0 0 0 RATE (3 : 0)

3.2.2.3.3 A write to the TLM.Stat register will initiate the automatic transmission of one complete Telemetry Word, as set forth in section 3.1.4.[9]

3.2.2.3.4 The telemetry output registers shall reset to all 0's on Power-On Reset.

3.2.2.4 Phase Registers

This section defines the Sample Counter and Bit Accumulation Counter phase registers. Reset states for the phase registers set the registers to their maximum possible value. This has been determined to be a more acceptable default than the alternatives.

3.2.2.4.1 Sample Counter Phase 0 (SCntPh0)

15 8	7 0
0 0 0 0 0 0 0 0	PHASE 0 LENGTH ARGUMENT

3.2.2.4.1.1 This register shall reset to $00FF on Power-On Reset.

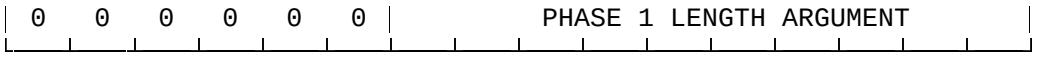
3.2.2.4.2 Sample Counter Phase 1 (SCntPhl)

15 8	7 0

[9] Software should attempt no further writes to TLM.Stat register until the output sequence is complete. With a nTLMclk of 64KHz and 32 bits of TLM data, start bit, stop bit, and dead bit between transmissions, this works out to

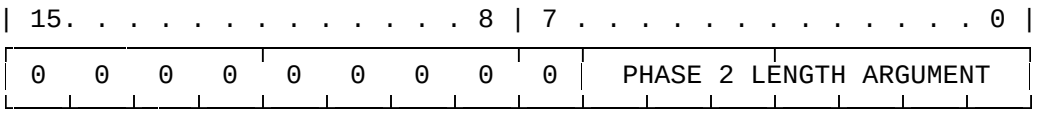
15.625 µs * 35 bits = 0.547 ms

per telemetry packet. Software must wait longer than this period before initiating another TLM transmit request.



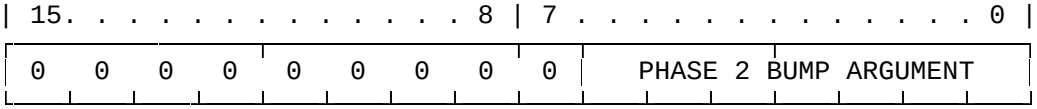
3.2.2.4.2.1 This register shall reset to $03FF on Power-On Reset.

3.2.2.4.3 Sample Counter Phase 2 (SCntPh2)



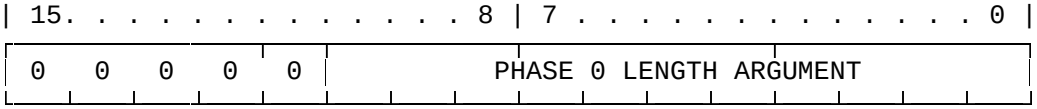
3.2.2.4.3.1 This register shall reset to $00FF on Power-On Reset.

3.2.2.4.4 Sample Counter Phase 2 Bump (SCntBump)



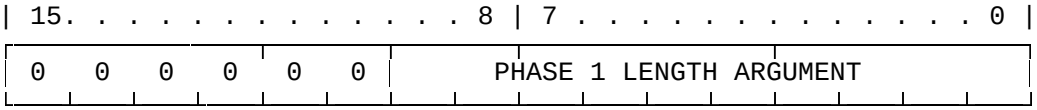
3.2.2.4.4.1 This register shall reset to $007F on Power-On Reset.

3.2.2.4.5 Bit Accumulation Counter Phase 0 (BACntPh0)



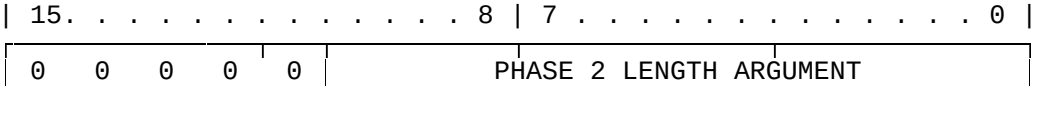
3.2.2.4.5.1 This register shall reset to $07FF on Power-On Reset.

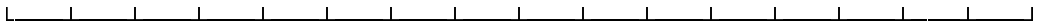
3.2.2.4.6 Bit Accumulation Counter Phase 1 (BACntPhl)



3.2.2.4.6.1 This register shall reset to $03FF on Power-On Reset.

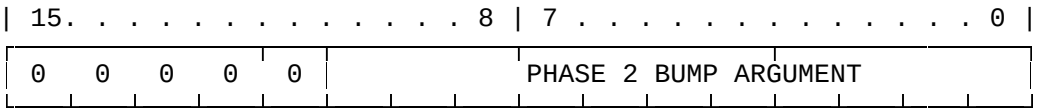
3.2.2.4.7 Bit Accumulation Counter Phase 2 (BACntPh2)





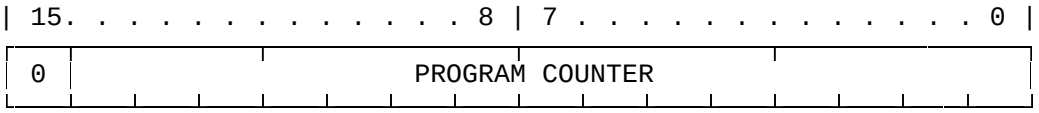
3.2.2.4.7.1 This register shall reset to $07FF on Power-On Reset.

3.2.2.4.8 Bit Accumulation Counter Phase 2 Bump (BACntBump)



3.2.2.4.8.1 This register shall reset to $07FF on Power-On Reset.

3.2.2.5 Program Counter (PC)



3.2.2.5.1 The Program Counter register contains the longword address of the currently-executing instruction. It is concatenated with an internal word pointer from the control logic to produce the word address into ROM.

3.2.2.5.2 The PC is incremented at the end of every instruction cycle to point at the next instruction. Since it can be treated as an arithmetic destination, dedicated relative branch and jump instructions are unnecessary. Relative branches are attained by simply adding the branch offset to the PC, and absolute jumps are simply a load of the PC.

Note, however, that if the PC is altered by an instruction, the next opcode fetch will actually occur from the resultant PC, plus 1. For example, MOV NF, T, PC, #$0000 would actually fetch from longword address $0001 rather than $0000 (ROM addresses $0002 and $0003 vs. $0000 and $0001).

3.2.2.5.3 Should the PC contain $7FFF when the increment occurs, it will wrap around to $0000.
In other words, a jump to $0000 can be accomplished by the MOV PC, #$7FFF instruction.

3.2.2.5.4 The reset state of the PC shall be such that the first instruction fetch will occur from address $0000.

3.2.3 Instruction Set

The CDU chip instruction set will consist of fixed-length 32-bit words, stored on natural word boundaries in the ROM. Field definitions within the word are chosen to be rigid, but may overlap, depending on the interpretation of the opcode[10] (see the following format descriptions).
The field definitions for the formats are:

1. OPCODE: 3-bit field describing arithmetic operation to be performed.

2. F: If 1, allows ALU operation to affect flags, else no effect.

[10] Format descriptions are developed with simplicity and uniformity in mind; as a consequence, there will be certain combinations of opcode and addressing modes that are nonsensical. This is understood at the outset.

3. N: Negation Bit. This value gets compared with the flag bit pointed to by TEST; the instruction gets executed if they are NOT equal.

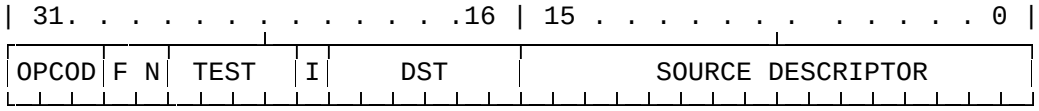
4. TEST: 4-bit field pointing to the flag bit to be used as the conditional test.

5. I: Immediate bit; 0 implies register addressing mode for source, 1 implies immediate mode.

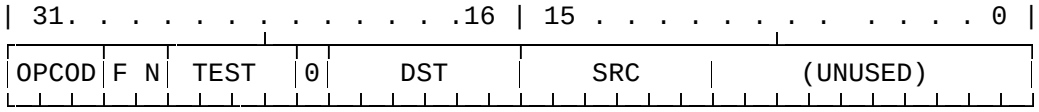
6. DST: Absolute 6-bit destination address, according to Operand Address Map (sec. 3.2.1.2)

7. SRC: Absolute 6-bit source register address, according to Operand Address Map (sec. 3.2.1.2). If indexed modes are added, this will be interpreted as the base register for the index.

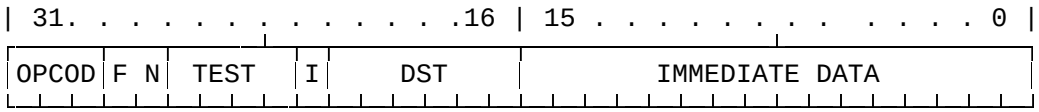
3.2.3.1 The general format for an instruction is:



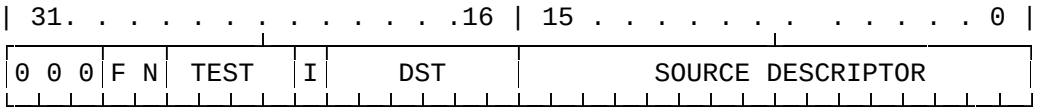
The format for register-mode instructions is then:



For typical immediate instructions:



MOV



Operation: DST = SRC

Flags (if F = 1):

C 0
V 0
S DST_{15}
Z $!(SRC_{15} \& SRC_{14} \& ... \& SRC_0)$

Notes:

This instruction, when the destination is the PC, is the conditional jump absolute instruction. For example, the 8086 instruction:

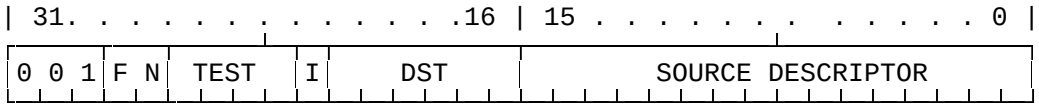
```
JC   #$0050                ; Jump to address $50 on carry set
```

becomes, in the CDU chip:

```
MOV    NF, C, PC, $004F    ; Load PC with ($50 - 1) if carry set.
```

Note the use of the NF argument to preserve the flags through the instruction for later tests.

ADD



Operation: DST = DST + SRC

Flags (if F = 1):

C $(DST + SRC)_{16}$

V $(!\ (DST_{14\ldots0} + SRC_{14\ldots0})_{15}\ \&\ DST_{15}\ \&\ SRC_{15})\ |$
$((DST_{14\ldots0} + SRC_{14\ldots0})_{15}\ \&\ !DST_{15}\ \&\ !SRC_{15})$

S $(DST + SRC)_{15}$

Z $!((DST + SRC)_{15}\ \&\ (DST + SRC)_{14}\ \&\ \ldots\ \&(DST + SRC)_{0})$

Notes:

Although an add-with-carry instruction does not exist, per se, in the CDU chip, it is not difficult to perform the same function. Where in the 8086 one might write:

```
MOV   A,VAL1L       ; Add low-order words
ADD   A,VAL2L
MOV   VAL1L,A
MOV   A,VAL1H       ; Add high-order words
ADC   A,VAL2H
MOV   VAL1H,A
```

to perform a 32-bit memory-memory add, the equivalent CDU code is:

```
ADD   F, T, VAL1L, VAL2L     ; Add low-order words
ADD   NF, C, VAL1H, $000l    ; If there was a carry out, increment high word of destination
ADD   F, T, VAL1H, VAL2H     ; Add high-order words
```

This instruction, when the destination is the PC, is the conditional jump relative instruction. For example:

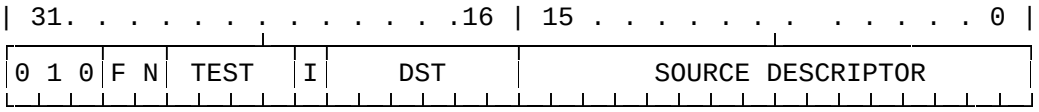
```
ADD   NF, C, PC, $0002       ; Skip the next two instructions without altering flags
```

This can implement a tight loop which is the equivalent of a WAIT instruction if we supply a 2's complement skip value to go backwards:

```
SUB   NF, A, PC, $FFFF       ; Loop to self until A is clear (accumulation complete)
```

Note that, in this example, the value $FFFF was used, rather than $0000; since the PC is automatically incremented at the end of each instruction, a value of $FFFF causes re-execution of the current instruction (a "branch to self"), while $0000 would simply point at the next instruction.

SUB



Operation: DST DST - SRC

Flags (if F = 1):

C $(DST + !SRC + 1)_{16}$
V $!(DST_{14..0} + !SRC_{14..0} + 1)_{15} \& DST_{15} \& !SRC_{15}) |$
$(DST_{14..0} + !SRC_{14..0} + 1)_{15} \& !DST_{15} \& SRC_{15})$
S $(DST + !SRC + 1)_{15}$
Z $!((DST + ! SRC + 1)_{15} \& (DST + ! SRC + 1)_{14} \& ... \&(DST + !SRC +1)_{0})$

Notes:

Although a subtract-with-carry instruction does not exist, per se, in the CDU chip, it is not difficult to perform the equivalent function. Where in the 8086 one might write:

```
MOV   A, VAL1L          ; Subtract low-order words
SUB   A, VAL2L
MOV   VAL1L, A
MOV   A, VAL1H          ; Subtract high-order words
SBB   A, VAL2H
MOV   VAL1H, A
```

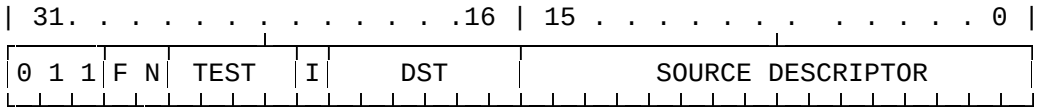
to perform a 32-bit memory-memory subtraction, the equivalent CDU code is:

```
SUB   F, T, VAL1L, VAL2L      ; Subtract low-order words
SUB   NF, NC, VAL1H, $0001    ; If there was no carry out (borrow), decrement high word of destination
SUB   F, T, VAL1L, VAL2L      ; Subtract high-order words
```

This instruction, when the destination is the PC, can also be used as a conditional jump relative instruction. Repeating an example from the ADD instruction, WAIT could be implemented as:

```
SUB   NF, A, PC, #$0001       ; Loop to self until A is clear (accumulation complete)
```

CMP



Operation: DST - SRC used to set flags (comparison)

Flags (if F = 1):

C $(DST + !SRC + 1)_{16}$

V $(!(DST_{14..0} + !SRC_{14..0} + 1)_{15} \& DST_{15} \& !SRC_{15}) \mid ((DST_{14..0} + !SRC_{14..0} + 1)_{15} \& !DST_{15} \& SRC_{15})$

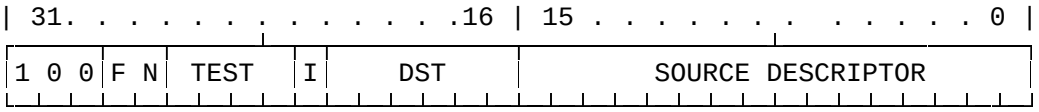
S $(DST + !SRC + 1)_{15}$

Z $!((DST + !SRC +1)_{15} \& (DST + !SRC +1)_{14} \& ... \&(DST + !SRC +1)_{0})$

Notes:

The results of the comparison are interpreted differently depending on whether signed or unsigned values are being compared!

AND

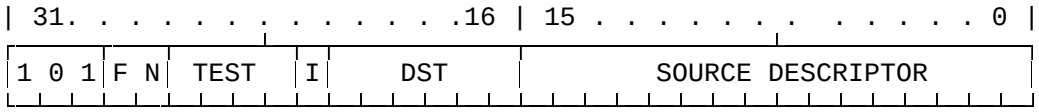


Operation: DST = DST & SRC (bitwise AND)

Flags (if F = 1):

C 0
V 0
S $(DST_{15} \& SRC_{15})$
Z $! ((DST \& SRC)_{15} \& (DST \& SRC)_{14} \& ... \&(DST \& SRC)_{0})$

OR

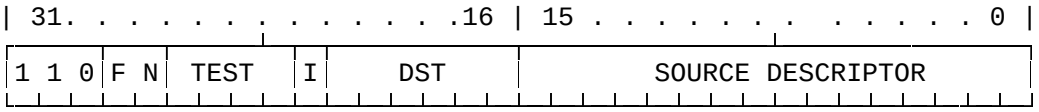


Operation: DST = DST | SRC (bitwise OR)

Flags (if F = 1):

C 0
V 0
S $(DST_{15} | SRC_{15})$
Z ! $((DST | SRC)_{15} \& (DST | SRC)_{14} \& ... \& (DST | SRC)_{0})$

XOR

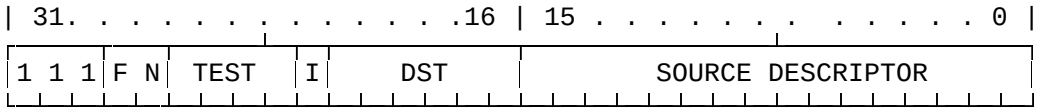


Operation: DST = DST ^ SRC (bitwise XOR)

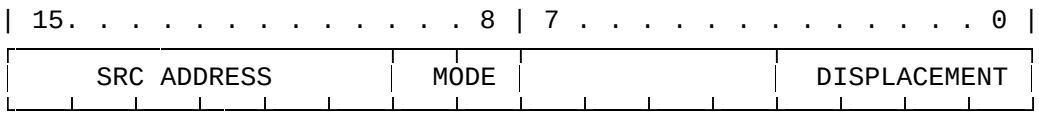
Flags (if F = 1):

C 0
V 0
S $(DST_{15} \wedge SRC_{15})$
Z $!((DST \wedge SRC)_{15} \& (DST \wedge SRC)_{14} \& ... \&(DST \wedge SRC)_{0})$

SHFT



Operation: Defined by contents of mode field in source descriptor, as follows:



Interpretation of the Source Descriptor fields differs from that of other opcodes. As with other opcodes, the I bit determines whether the source descriptor is to be interpreted as an immediate value or an indirect reference to a source address; however, in this case the only displacement field will be accessed from a memory location. In other words, if I is 0, the SRC address field of the SHFT instruction is used to get the Displacement value from memory, while the Displacement field in the opcode source descriptor is ignored. Conversely, if I is 1, the Displacement is taken from the opcode's Displacement field, and the SRC address field is unused. In either case, the Mode is taken from the Mode field of the Source descriptor field of the opcode. Or, in yet other words, the Mode field is forced to be Immediate, regardless of the state of the I bit.

The mode field is defined as follows:

Mode	Operation		
00	Arithmetic Shift Right	i > 15-Disp:	$DST_i = DST_{15}$
		else:	$DST_i = DST_{(i+Disp)}$
01	Logical Shift Right	i > 15-Disp:	$DST_i = 0$
		else:	$DST_i = DST_{(i+Disp)}$
10	Arithmetic/Logical Shift Left	i < Disp:	$DST_i = 0$
		else:	$DST_i = DST_{(i-Disp)}$
11	Rotate Left		$DST_{((i+Disp) \bmod 16)} = DST_i$

And the Displacement field is the number of bit positions right or left that the destination value gets shifted, from 0 to 15. A displacement value of 0 results in no effective shift.

Flags (if F = 1):

C 0
V 0
S $SHFT(DST)_{15}$
Z $SHFT(DST)_{15}$ & $SHIFT(DST)_{14}$ & ... &$SHFT(DST)_0$

3.3 External ROM Read Cycle

Instruction reads from the external ROM occur in the first four internal T-states of the CDU ASIC instruction cycle, as diagrammed below. In normal operation, IAddr(15:1) and IAEn will remain at their last values from T3 to the next T0. The instruction words are latched in the vicinity of the rising edges of Clk that end states T1 and T3.

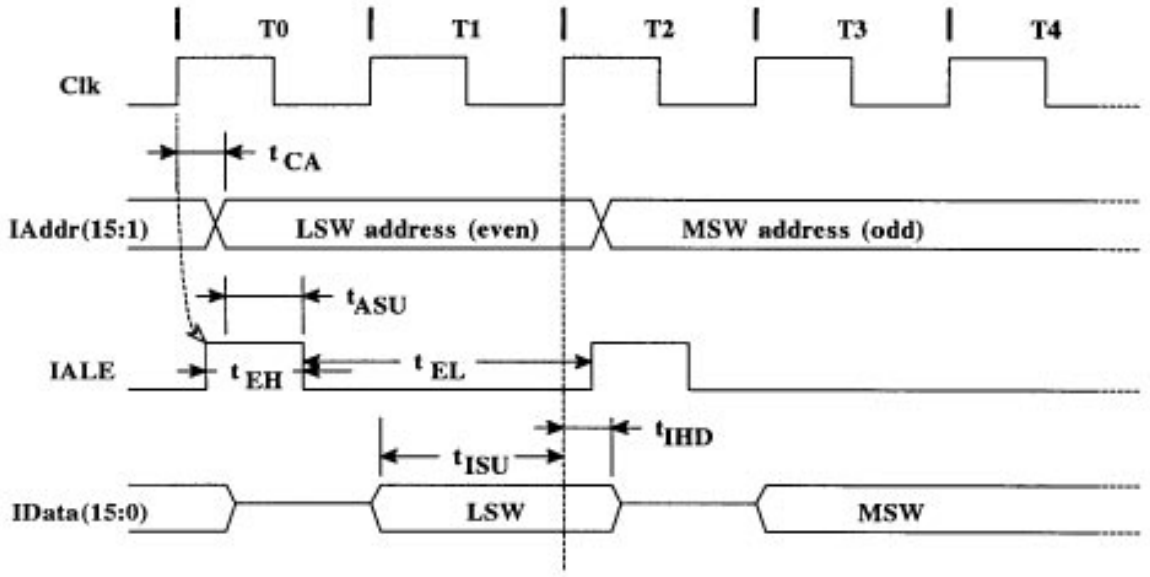


Figure 3-1. Typical Read Cycle Timing Diagram

	Min (ns)	Max (ns)
tEH	(0.5*tcyc - 20)	(0.5*tcyc + 20)
tEL	(1.5*tcyc- 20)	
tCA		20
tASU	30	
tISU	35	
tIHD	10	

Table 3-2. ROM Access Timing Specifications (Preliminary) [11]

[11] For information only! For manufacturer guaranteed timing refer to table 5-5.

3.4 Analog-to-Digital Conversion Interface

The timing diagram here is based on the Analog Devices AD7572A converter interface requirements. The CDU ASIC side of the interface is governed by the rising edge of Clk, and that of the ADC by the falling edge of ADClk. This specification gives a 2.048 MHz ADClk from an 8.192 MHz Clk at a nominal 50% duty cycle. Tolerances are named to keep the duty cycle well within the 40%-60% limits allowed by the converter.

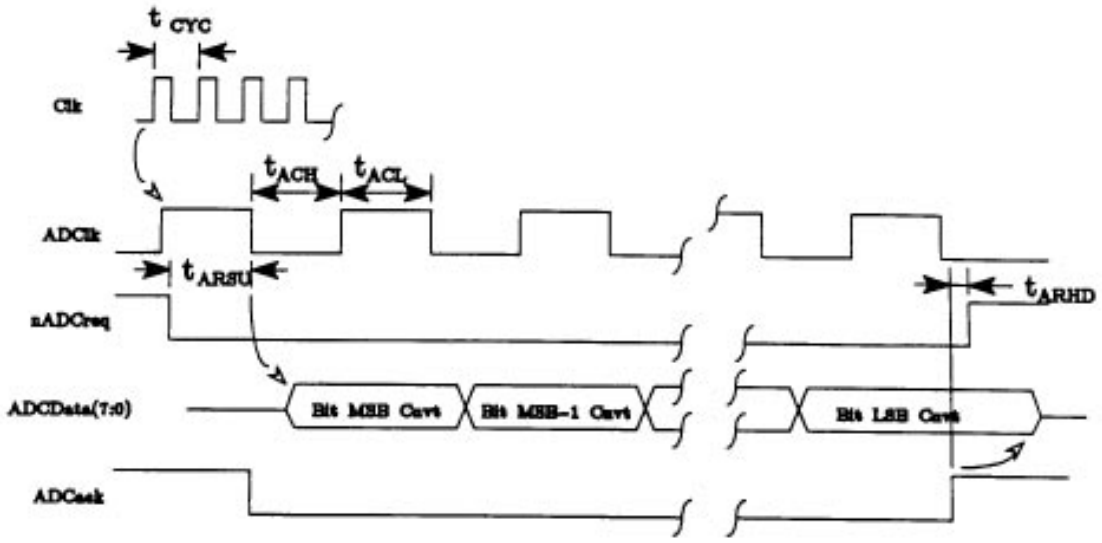


Figure 3-2. Essential Timing of ADC Interface

	Min (ns)	Max (ns)
tACH	(2*tcyc - 20)	(2*tcyc + 20)
tACL	(2*tcyc - 20)	(2*tcyc + 20)
tARSU	tcyc - 40	(4*tcyc - 40)
tARHD	tcyc	9*tcyc

Table 3-3. ADC Timing Specifications (Preliminary) [12]

[12] For information only! For manufacturer guaranteed timing refer to table 5-5.

4. TESTABILITY CONSTRUCTS

4.1 Honeywell-Recommended Constructs

4.1.1 The CDU chip will support Boundary Scan of all pins of the chip, where meaningful.[13,14]

4.1.2 Internal scan paths will be used to the extent that Honeywell's Rigel test pattern generator tool can automate fault test vector generation; however, storage nodes not in the scan path are allowed, if they can be observed and controlled through some other direct means.[15]

4.1.3 If transparent latches are required to fulfill throughput requirements, they will be surrounded by the Honeywell-recommended logic to make them testable under Rigel control; this logic makes the latch look like combinatorial logic when in test mode.

4.1.4 The design will use the Honeywell On-Chip Monitor (OCM) macrocell for the control of internal scan paths. Use of the OCM provides the IEEE 1149.1 serial interface to the scan paths. Self-test logic options for the OCM will not be selected, nor any other options that unnecessarily increase gate count above the minimum required for basic testability within a chip test fixture.

4.1.5 The Command Detector Unit does not make use of the IEEE 1149.1 interface; however, the interface on the ASIC would be applicable to future flight applications that require Built-In Self-Test (BIST).

4.2 JPL Test and Debugging Extensions

The Debug pin of the CDU ASIC makes available modes of operation that are useful for hardware test and for debugging the microcode. The Debug mode is very simple in implementation, but with powerful implications.

Due to the potentially catastrophic side effects of the Debug mode, the Debug pin is NOT available for any flight application! It is intended for non-flight development articles only!

4.2.1 The Debug pin directly forces the CDU ASIC into the Debug mode; it may be applied anywhere within the instruction, but for most purposes, it should change state on rising edges of the clock.

The Debug pin forces the following effects:

1. The 16 output pins IAddr(15:1) and IALE change mode. They will act as a 16-bit output bus that monitors the output value from the internal ALU. The meaning of the observed values is implementation-specific, and is to be documented in the CDU ASIC Design Book.

2. While Debug is active, increments of the PC are suppressed.

---

[13] Note that we have no board-level testability requirements that force us to include boundary scan, but it can make it easier to test a CDU module without probing the board directly, if we are allowed to define our own test connector that is independent of in-spacecraft restrictions that effect Support Equipment, Telemetry, etc.

[14] It is possible that Power-On Reset cannot be in the boundary scan path, and will require specific test vector sets to achieve full testability.

[15] On-chip RAM is an example of storage that is not directly in the scan path, and therefore is not testable by Rigel; we require that it be accessible in a way that allows the required level of fault coverage.

3. While Debug is active, the ALU-specific flags in the FLAGS word may be directly written by the executing instruction. Normally, the ALU state has priority. See section 3.2.2.2.

Other functions in the CDU ASIC will proceed normally.

4.2.2 In the Debug mode, normal microcode flow has been suspended by suppressing the PC increment. If the normal external ROM (or ROM emulator) is disabled (high-impedance) when Debug pin is active, an alternative instruction may be forced onto the CDU ASIC data bus by other, non-flight, external hardware. A MOV instruction, for instance, might be forced for the purposes of presenting a RAM location on the IAddr(15:1) and IALE pins. This facility can be used for improved efficiency in performing RAM functional verification.

4.2.3 The operation described in TBSL has applicability to microcode debug. "Stealing" an instruction cycle from the CDU microcode is possible with the Debug pin to perform examines, deposits, and execution; these form the critical components of a complete software development system. Note that this capability is not required by the CDU design team, but is a powerful fallback capability should sticky code-related problems arise in the prototyping phase.

4.2.4 The behavior of the FLAGS word is slightly altered in the Debug mode, to allow a software developer easier access to the ALU flags. A side effect is that any potential software development system constructed for the CDU ASIC cannot attempt to run an actual debug monitor on the ASIC.[16] Any software development system must contain its own rather substantial intelligence, if it is to emulate the functionality of a standard In-Circuit Emulator.

[16] The design of In-Circuit Emulators, debug monitors, and related devices for supporting software development is a highly specialized and complex subject that cannot be dealt with in this specification. This section refers to limitations of scope that are critical to the expert designer of such systems.

5. ELECTRICAL CHARACTERISTICS

5.1 Electrical Test Requirements

Test	Subgroups (Per MIL-STD-883, Method 5005, Table 1)
Initial (Pre Burn-In)	1,7
Interim (Post Static I Burn-In)	1*,7*
Delta Calculations*,**	
Interim (Post Static II Burn-In)	1*,7*
Delta Calculations*,**	
Final (Post Dynamic Burn-In)	1*,2,3,7*,8,9,10,11
Delta Calculations*,**	
Group A	1,2,3,7,8,9,10,11
Group C End Point electrical***	1,2,3,7,8,9,10,11
Delta Calculations**	

Table 5-1. Electrical Test Requirements

* PDA applies to these subgroups.
** Deltas shall be calculated relative to the initial electrical parameters. Delta limits of table 5-8 herein shall apply.
*** Group C Lifetest shall be performed using the dynamic burn-in configuration of table 5-7 herein.

5.2 Absolute Maximum Ratings[17,18,19]

Symbol	Parameter	Ratings		Units
		Min.	Max.	
V_{DD}	Supply Voltage	-0.5	7.0	V
V_{IN}	Input Voltage Range	GND - 0.5	V_{DD} + 0.5	V
I_{OUT}	DC or max. Output Current		50	mA
P_D	Max. Package Power Dissipation		4	W
T_{ST}	Storage Temperature Range	-65	150	^oC
T_S	Lead Temperature (Soldering, 5s)		270	^oC
T_J	Junction Temperature		175	^oC
Θ_{JC}	Thermal Resistance, Junction to Case		4	$^oC/W$
V_{ESD}	ESD Protection Voltage - Class 2 (MIL-STD-883, Method 3015)	2000		V

Table 5-2. Absolute Maximum Ratings

5.3 Recommended Operating Conditions[17,20]

Symbol	Parameter	Ratings		Units
		Min.	Max.	
V_{DD}	Supply Voltage	4.5	5.5	V
T_A	Ambient Temperature	-55	125	^oC
f_{max}	Max. Operating Frequency		8.2	MHz
t_r, t_f	Input Rise Time, Input Fall Time		500	ns

Table 5-3. Recommended Operating Conditions

---

17 Stresses above the absolute maximum rating may cause permanent damage to the device. Extended operation at the maximum levels may degrade performance and effect reliability.

18 Values are guaranteed but not tested.

19 $-55^oC \le T_c \le 125^oC$ except as noted.

20 Extended operation outside recommended limits may degrade performance and effect reliability.

5.4 DC Characteristics

5.4.1 DC Electrical Performance Characteristics

Parameter	Symbol	Test Condition	Subgroup	Limit		Unit
				Min	Max	
Input Threshold Voltage	V_{IH} [21]	V_{DD}=5.5V	1,2,3		3.85	V
	V_{IL} [21]	V_{DD}=4.5V	1,2,3	1.35		V
	V_{T+} [22]	V_{DD}=5.5V	1,2,3	2.5	3.7	V
	V_{T-} [22]	V_{DD}=4.5V	1,2,3	1.4	2.4	V
	V_{HYS} [22]	V_{DD}=5.5V	1,2,3	0.6	1.7	V
Input Leakage Current	I_{IH1} [23]	V_{DD}=5.5V, V_{IH}=V_{DD}	1,2,3	-10	10	µA
	I_{IH2} [24]	V_{DD}=5.5V, V_{IH}=V_{DD}	1,2,3	-10	10	µA
	I_{IH3} [25]	V_{DD}=5.5V, V_{IH}=V_{DD}	1,2,3	50	550	µA
	I_{IL1} [23]	V_{DD}=5.5V, V_{IL}=GND	1,2,3	-10	10	µA
	I_{IL2} [24]	V_{DD}=5.5V, V_{IL}=GND	1,2,3	-550	-50	µA
	I_{IL3} [25]	V_{DD}=5.5V, V_{IL}=GND	1,2,3	-10	10	µA
Output Leakage Current	I_{OZH}	V_{DD}=5.5V, V_O=VDD	1,2,3	-10	10	µA
(Tristate)	I_{OZL}	V_{DD}=5.5V, V_O=GND	1,2,3	-10	10	µA
Output Voltage	V_{OH1} [26]	V_{DD}=4.5V, I_{OH}=-3mA	1,2,3	4.0		V
	V_{OH2} [27]	V_{DD}=4.5V, I_{OH}=-9mA	1,2,3	4.0		V
	V_{OL1} [26]	V_{DD}=4.5V, I_{OL}=3mA	1,2,3		0.5	V
	V_{OL2} [27]	V_{DD}=4.5V, I_{OL}=9mA	1,2,3		0.5	V
Output Current	I_{OH1} [26]	V_{DD}=4.5V, V_{OH}=4V	1,2,3		-3	mA
	I_{OH2} [27]	V_{DD}=4.5V, V_{OH}=4V	1,2,3		-9	mA
	I_{OL1} [26]	V_{DD}=4.5V, V_{OL}=0.5V	1,2,3	3		mA
	I_{OL2} [27]	V_{DD}=4.5V, V_{OL}=0.5V	1,2,3	9		mA

Table 5-4. DC Performance Characteristics

Parameter	Symbol	Test Condition	Subgroup	Limit	Unit

[21] All Inputs, except: nPOR0, nPOR1, TRSTN
[22] Schmitt Trigger Inputs: nPOR0, nPOR1, TRSTN
[23] All Inputs, except: TestClk, Rigel, TDI, TMS, nPOR0, nPOR1, nBitRate0, nBitRate1
[24] Inputs with Pull-Ups: nPOR0, nPOR1, nBitRate0, nBitRate1, TDI, TMS
[25] Inputs with Pull-Downs: Rigel, TestClk
[26] 3mA Outputs: AGC_(9:0), ADClk, nADCreq, nCmdData, nCmdClk, nLock, nLock1, nTLMData, nTLMClk
[27] 9mA Outputs: IAddr(15:1), IALE

				Min	Max	
Standby Supply Current	I_{DDSB}	V_{DD}=5.5V, V_{IN}=V_{DD} or GND, Fc=0 Hz	1,2,3		800	µA
Quiescent Current	I_{DDQ}	V_{DD}=5.5V, V_{IN}=V_{DD} or GND, Fc=0 Hz	1,2,3		5	µA
Operating Current	I_{DDOP}[28]	V_{DD}=5.5V, V_{IN}=V_{DD} or GND, Fc=8.2 MHz	1,2,3		100	mA
Input Capacitance [29]	C_{IN}				15	pF
Output Capacitance [29,30]	C_{OUT}				15	pF

Table 5-4. DC Performance Characteristics (Cont'd)

5.4.2 Estimated Power Dissipation

$P_{Total} = P_{Internal} + P_{Outputs} + P_{Quiescent} = 258.8 + 22.3 + 4.4 = 285.5$ mW

$P_{Internal} = 258.8$ mW [31]

$P_{Outputs} = 22.3$ mW [31]

$P_{Quiescent} = V_{DD} * I_{DDSB} = 4.4$ mW

Where: $V_{DD} = 5.5$V
$I_{DDSB} = 800$ µA

5.4.3 IDDQ Testing

Quiescent Current (IDDQ) testing shall be accomplished by using the Stuck-at Fault test vectors generated with Rigel, or a subset thereof, plus independently developed vectors, as determined by JPL. Measurements shall be taken at every vector, unless otherwise indicated, recorded and compared to the IDDQ limit. The following statistical values shall be provided: Minimum, Maximum, Mean. The IDDQ limits shall be established by JPL after characterization of Engineering Model parts fabricated from the same wafer lot as the flight parts.

---

[28] This value is based on the supplied functional test vector set and the ANDO tester loading (85 pF) and may not be applicable to system operation.
[29] Guaranteed but not tested.
[30] Refers to internal capacitance.
[31] Per CDU Gate Array Verify Report, dated 93/02/18

5.4.4 Pull-Down Resistors

Certain input pins are required to be held to a logic low state in flight. The internal pull-down resistor design option shall be used on the following input pads to assure a low state:

Rigel
TestClk

5.4.5 Pull-Up Resistors

Certain input pins are required to be held to a logic high state in flight. The internal pull-up resistor design option shall be used on the following input pads to assure a high state:

nPOR0, nPOR1
nBitRate0, nBitRatel
TDI
TMS

5.5 AC Characteristics

5.5.1 AC Electrical Performance Characteristics

Parameter	Symbol	Test Condition $V_{IN}=V_{DD}$ or GND	Subgroup	Specification Limit		Tester Limit		Unit
				Min	Max	Min	Max	
Functional Tests		V_{DD}=4.5 & 5.5V Fc=8.2 MHz	7,8	pass		pass		
Propagation Delay:		V_{DD}=4.5 & 5.5V Fc=500 kHz	9,10,11					
Clk ↑ to IALE ↑	T_{CHEH}		9,10,11	4.0	33.0	4.0	30.2	ns
Clk ↓ to IALE ↓	T_{CLEL}		9,10,11		37.0		33.6	ns
IALE Pulse Width[32]	T_{EHEL}		9,10,11	56.0	64.0			ns
IALE - Clk Delta	T_{DEL}		9,10,11	-5.0	3.0	-5.0	3.0	ns
Setup Time:		V_{DD}=4.5 & 5.5V Fc=500 kHz	9,10,11					
IDATA(15:0) to Clk ↑	T_{ISU}		9,10,11		3.0		3.0	ns
Hold Time:		V_{DD}=4.5 & 5.5V Fc=500 kHz	9,10,11					
Clk ↑ to IDATA(15:0)	T_{IHD}		9,10,11		13.0		13.0	ns

Table 5-5. AC Performance Characteristics

5.5.2 AC Timing Diagram

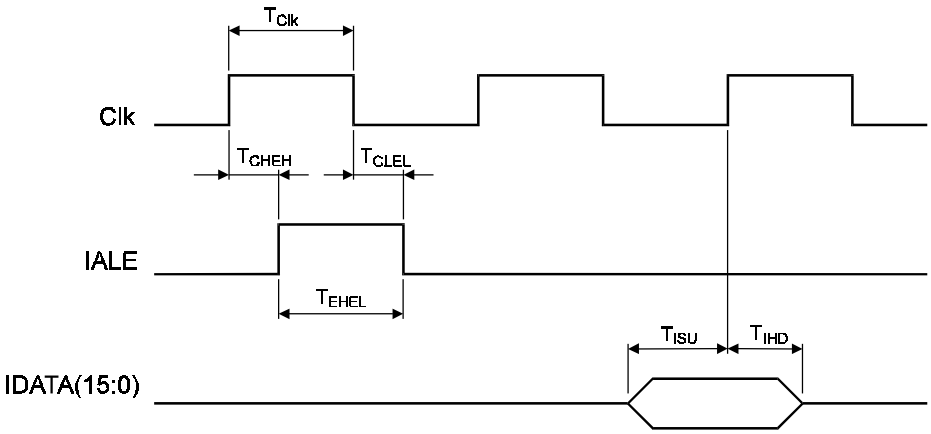


Figure 5-1. AC Timing Diagram

[32] Indirectly measured by $T_{DEL}=T_{EHEL}-T_{Clk}=T_{CLEL}-T_{CHEH}$; $T_{Clk}= 1/2 *1/ f_{Clk}$

5.5.3 Timing Analysis

Pre-layout and post-layout timing shall pass the QuickSim simulation of the vectors supplied to the contractor without setup/hold timing violations.

5.5.3.1 Pre-Layout Timing Margins

Pre-layout timing margins shall be calculated by using standard extreme-value analysis. The extreme values for the cell library shall be supplied by the contractor. Critical paths will be identified and margin calculated via Mentor or Honeywell software toolsets, or a combination thereof.

5.5.3.2 Post-Layout Timing Margins

Post-layout analysis of the device shall show positive margin on internal critical paths over all operating conditions. The analysis will follow the same form as the pre-layout analysis, with the post-layout timing values annotated to the design file by the contractor.

5.5.3.3 Tester Specification Limits

Tester Specification limits (max. propagation delays) in table 5-5 have been adjusted for modified output levels and for differences in output loading in the ANDO tester environment. Modified output levels are required to account for impedance mismatches between device outputs and the ANDO tester environment. The level at which an output is considered to have switched has been changed from 50% of VDD to 1V for low to high transitions and VDD-0.5V for high to low transitions for the 3mA buffer and to 1V for low to high transitions and VDD-1V for high to low transitions for the higher drive buffers (6mA, 9mA, 12mA, 15mA).

$t_{SPEC}(\text{Tester}) = t_{SPEC}(\text{System}) - (T_{Offset\_fixed} + \text{CLoad} * \text{LoadingFactor})$

Where:

C_{Load} = 50 pF (System Load)

For 3 mA drive buffer:

Low to High transition:	T_{Offset_fixed} = -2.0 ns	LoadingFactor = 0.205
High to Low transition:	T_{Offset_fixed} = -1.6 ns	LoadingFactor = 0.241

For 9 mA drive buffer:

Low to High transition:	T_{Offset_fixed} = -0.6 ns	LoadingFactor = 0.069
High to Low transition:	T_{Offset_fixed} = -0.6 ns	LoadingFactor = 0.081

5.5.5 Tester Load Circuit

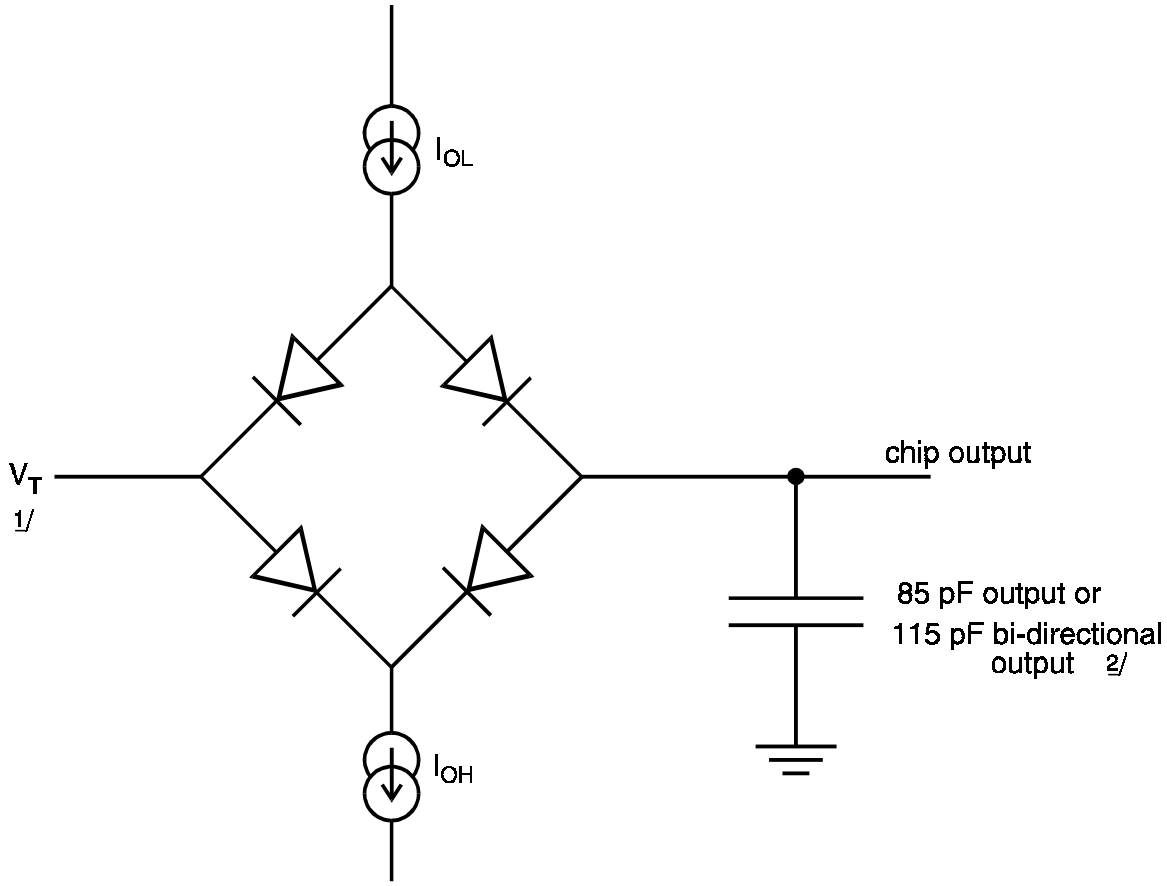


Figure 5-2. Tester Load Circuit

1/ V_T is a variable dependent upon the test parameter.
2/ This capacitance is actually partially distributed through the fixturing so that the device is actually loaded by a transmission line.

5.6 Burn-In

5.6.1 Static Burn-In

The Static Burn-In conditions shall be as specified in JPL General Specification CS515837, Rev. B. The burn-in configuration shall be as shown in table 5-7.

5.6.2 Dynamic Burn-In

The Dynamic Burn-In conditions shall be as specified in JPL General Specification CS515837, Rev. B. The burn-in configuration shall be as shown in table 5-7. Input stimuli (STIM) to exercise the device shall be applied by using the RIGEL Stuck-at Fault test vectors, or a subset thereof, plus independently developed test vectors, as determined by JPL. At least one output (MON) shall be monitored during burn-in to assure that the output is toggled and the circuit functioning.

5.6.3 Burn-In Conditions

	Static I	**Static II**	**Dynamic**	**QCI - Life Test**
Duration	48 hours	48 hours	240 hours	2000 hours
Voltage	6.5 V	6.5 V	6.5 V	6.0 V
+ Tolerance	+0.1V	+0.1V	+0.1V *	+0.1V *
- Tolerance	-0.25V	-0.25V	-0.25V	-0.25V
Temperature	125 ^{O}C	125 ^{O}C	125 ^{O}C	125 ^{O}C
+ Tolerance	+5 ^{O}C	+5 ^{O}C	+5 ^{O}C	+5 ^{O}C
- Tolerance	-0 ^{O}C	-0 ^{O}C	-0 ^{O}C	-0 ^{O}C

Table 5-6. Burn-In Conditions

* Applies to Average Power Supply Voltage. Tolerance for Dynamic Switching Noise is +0.25V.

5.6.4 Burn-In Configuration

Signal Name	Pin No.[33]	Type[34]	Burn-In Test Connection[35]			Description
			Static I	Static II	Dynamic	
nPOR0, nPOR1	31, 34	SINPU	GND	VDD	STIM[36]	Master Power-On Resets.
Clk	24	IN	GND	VDD	STIM[36]	Master Digital Clock
IAddr(15:1)	66 - 73, 76 - 82	OUT9	GND	VDD	VDD	Instruction Address
IALE	65	OUT9	GND	VDD	VDD	Instruction Address Latch Enable
IData(15:0)	62 - 55, 52 - 45	IN	GND	VDD	STIM[36]	Instruction Data.
AGC (9:0)	14 - 23	OUT3	GND	VDD	VDD	Automatic Gain Control output
ADClk	3	OUT3	GND	VDD	VDD	Analog-to-Digital conversion Clock
nADCreq	2	OUT3	GND	VDD	VDD	Analog-to-Digital Conversion request
ADCack	83	IN	GND	VDD	STIM[36]	Analog-to-Digital Conversion acknowledge
ADCData(7:0)	13, 10 - 4	IN	GND	VDD	STIM[36]	Analog-to-Digital Converter Data
nBitRate0 nBitRate1	37 38	INPU	GND	VDD	STIM[36]	Bit Rate sample descriptor
nCmdData	41	OUT3	GND	VDD	VDD	Command Data Bit output
nCmdClk	42	OUT3	GND	VDD	VDD	Command Data Bit Clock
nLock	36	OUT3	GND	VDD	VDD	Subcarrier Lock Status
nLockl	35	OUT3	GND	VDD	VDD	Subcarrier Lockl Status
nTLMData	39	OUT3	GND	VDD	VDD	Telemetry Output Data
nTLMClk	40	OUT3	GND	VDD	VDD	Telemetry Output Clock
STIM[36]	25	INPD	GND	VDD	STIM[36]	Rigel test command
TestClk	26	INPD	GND	VDD	STIM[36]	Test Circuitry Clock
TDI	28	INPU	GND	VDD	STIM[36]	Test Data In
TMS	27	INPU	GND	VDD	STIM[36]	Test Mode Select
TRSTN	30	SIN	GND	VDD	STIM[36]	Test Reset
TDO	29	TRI3	GND	VDD	MON[37]	Test Output Data
Debug	63	IN	GND	VDD	STIM[36]	Debug Enable
VddP(5:0)	75, 53, 44, 33, 11, 1	VDD	6.5V	6.5V	6.5V (6V for Lifetest)	Vdd Power Pads.
VssP(5:0)	74, 54, 43, 32, 12, 84	VSS	0.0V	0.0V	0.0V	Vss Power Pads.

Table 5-7. Burn-In Configuration

[33] The pin numbers of bus signals are in the same sequence as the signals.
[34] For a description of the signal type refer to table 5-9.
[35] All inputs and outputs shall be tied to the specified voltage level through a 2.2 kΩ resistor (±5%, 1/4W).
[36] Stimulated Inputs
[37] Monitored Output

5.6.5 Delta Limits

Symbol	Parameter	Spec. Limits		Units	Delta Limits	Units
		Min	Max			
I_{DDSB}	Static Supply Current		800	µA	80	µA
I_{IH1}	Input Leakage Current: Inputs without Pull-Ups or Pull-Downs	-10	10	µA	±1	µA
I_{IH2}	Input Leakage Current: Inputs with Pull-Ups	-10	10	mA	±1	µA
I_{IH3}	Input Leakage Current: Inputs with Pull-Downs	50	550	mA	±55	µA
I_{IL1}	Input Leakage Current: Inputs without Pull-Ups or Pull-Downs	-10	10	µA	±1	µA
I_{IL2}	Input Leakage Current: Inputs with Pull-Ups	-550	-50	µA	±55	µA
I_{IL3}	Input Leakage Current: Inputs with Pull-Downs	-10	10	µA	±1	µA
$I_{OZL,}$ I_{OZH}	Output Leakage Current: Tri-states	-10	10	µA	±1	µA
I_{OL1}, I_{OH1}	Output Current: 3 mA Outputs	-3	3	mA	±300	µA
I_{OL2}, I_{OH2}	Output Current: 9 mA Outputs	-9	9	mA	±900	µA

Table 5-8. Delta Limits

5.7 Pin Type Description

Pin Name		Drive	I/O	Type
IN	=		CMOS,	Input Signal
INPU	=		CMOS,	Input Signal with Pull-Up
INPD	=		CMOS,	Input Signal with Pull-Down
TRI3	=	3mA,	CMOS,	Tri-State Signal
OUT3	=	3mA,	CMOS,	Output Signal
OUT9	=	9mA,	CMOS,	Output Signal
SIN	=		TTL,	Schmitt Trigger Input Signal
SINPU	=		TTL,	Schmitt Trigger Input Signal with Pull-Up

Table 5-9. Pin Type Description

6. PHYSICAL CHARACTERISTICS

6.1 Pin Assignment

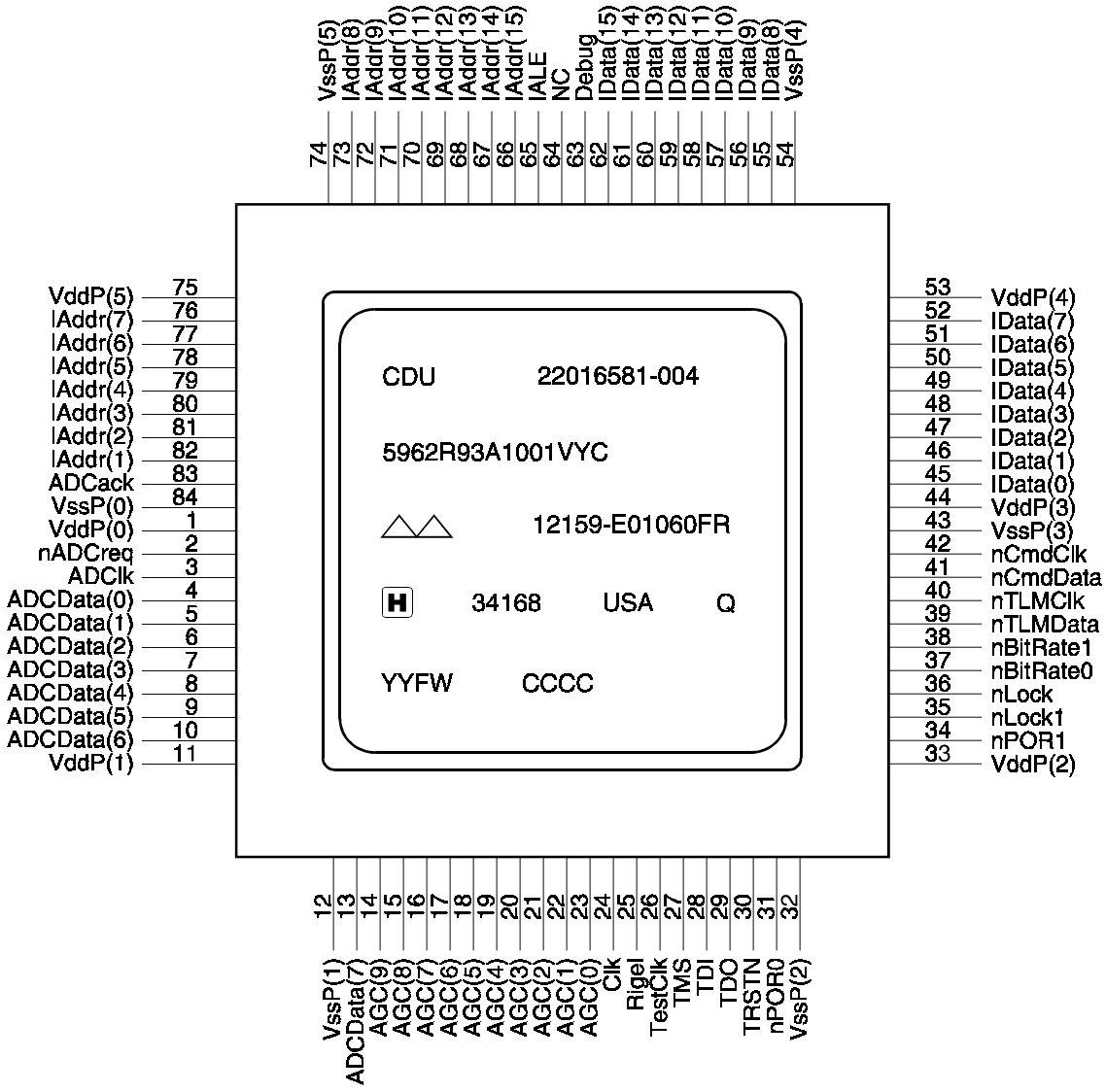


Figure 6-1. CDU ASIC Pinout Assignments

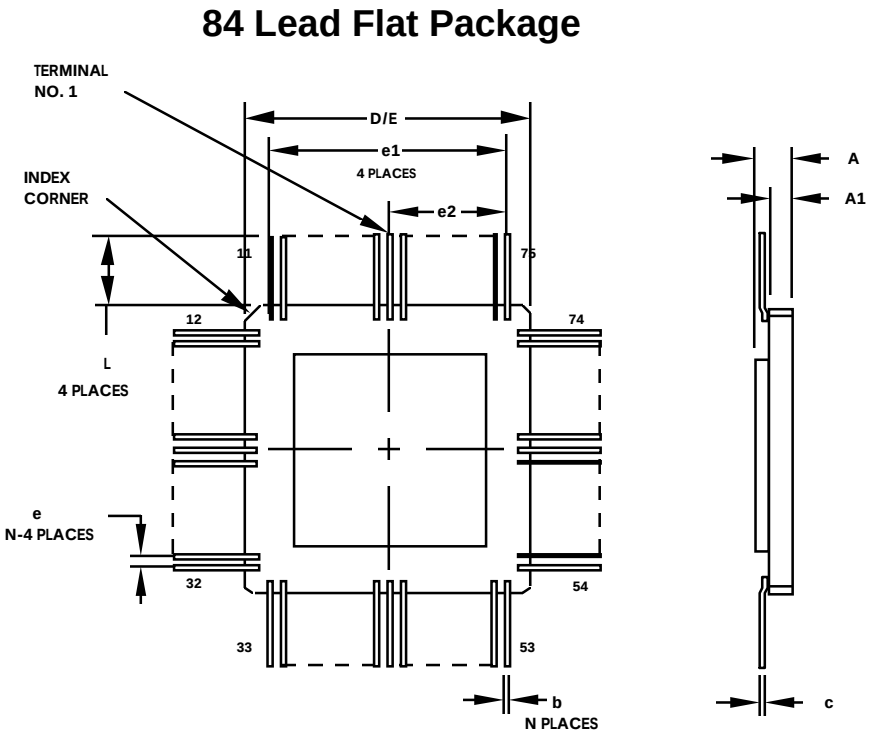
6.2 Bonding Diagram

Die Cavity: 520 mil/side
VDD Plane: pins 11, 33, 53, 75
VSS Plane: pins 12, 32, 54, 74
Die Attach: VSS
Lid and Seal Ring: VSS

Figure 6-2. Bonding Diagram (84-pin Flatpack)

6.3 Package Outline

The design shall be compatible with package styles approved by JPL for flight parts, i.e., hermetically sealed ceramic flatpacks, with strain-relieving lead bends.



PACKAGE DIMENSIONS			
Symbol	Dimensions in inches		
	min		max
A	0.095		0.125
A1	-		0.105
b	0.014		0.018
c	0.006		0.008
D/E	1.135		1.165
e		0.050 BSC	
e1		1.000 BSC	
e2		0.500 BSC	
L	0.350		-
N		84	

Figure 6-3. Package Outline (84-pin Flatpack)

6.4 Marking Diagram

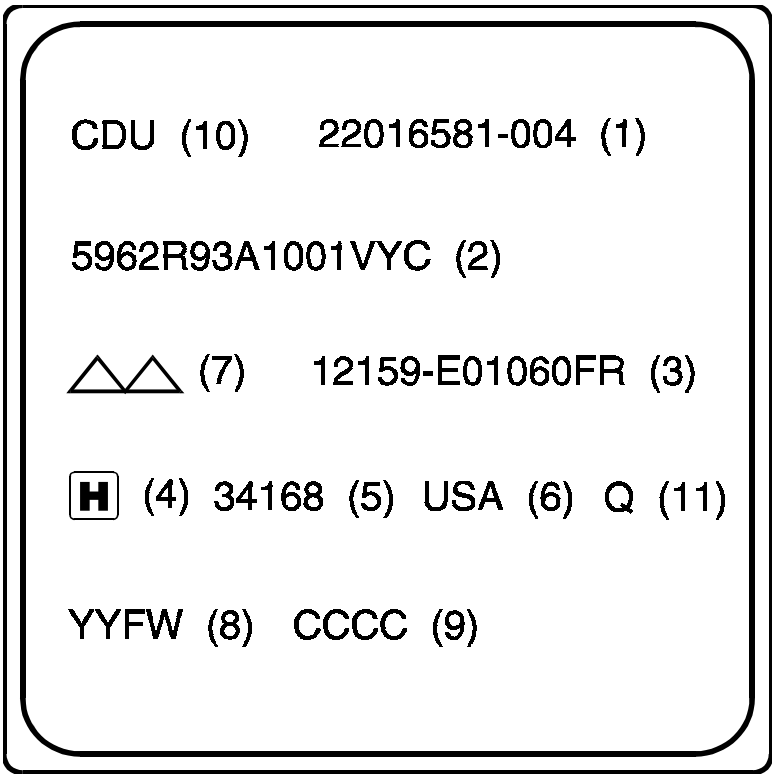


Figure 6-4. Marking Diagram

(1) Honeywell Part Number
(2) QML Number (Flight Units only)
(3) Customer Part Number
(4) Honeywell Trademark
(5) Federal Supplier Manufacturing Number
(6) Country of Origin
(7) Pin 1 indicator and ESD identifier
(8) Date Code -Year and Fiscal Week of Lid Seal.
YY = Year
FW = Fiscal Week
(9) Serialization (Traceability Capability to Die)
(10) Chip Name (If required)
(11) QML Mark (Flight Units only)

Filename: ST12159.C
Directory: H:\USERS\514\SPECS\ACT-DETL
Template: F:\USERS\JSANSONE\MSOFFICE\WINWORD\TEMPLATE\NORMAL.DOT
Title: CDU Spec scanned
Subject:
Author: MIKE SANDOR
Keywords:
Comments:
Creation Date: 05/10/94 2:04 PM
Revision Number: 5
Last Saved On: 05/20/94 4:40 PM
Last Saved By: Hubert Schafzahl
Total Editing Time: 145 Minutes
Last Printed On: 08/08/95 2:40 PM
As of Last Complete Printing
Number of Pages: 52
Number of Words: 25,554 (approx.)
Number of Characters: 145,663 (approx.)